

Spring 2008 22C:021:A01/A03

Computer Science II : Data Structures

Instructor: Wenli He (wenli-he@uiowa.edu)

Supervisor: Jim Cremer (cremer@cs.uiowa.edu)

Department of Computer Science

University of Iowa

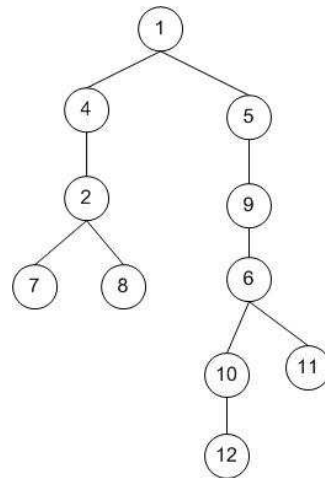
02/28/2008

Tree Traversal

- Implementation of Trees and Binary Trees
- Binary Search Tree
- Preorder, inorder and postorder traversal
- level-order traversal
- Solution to Quiz3 (by Justin)

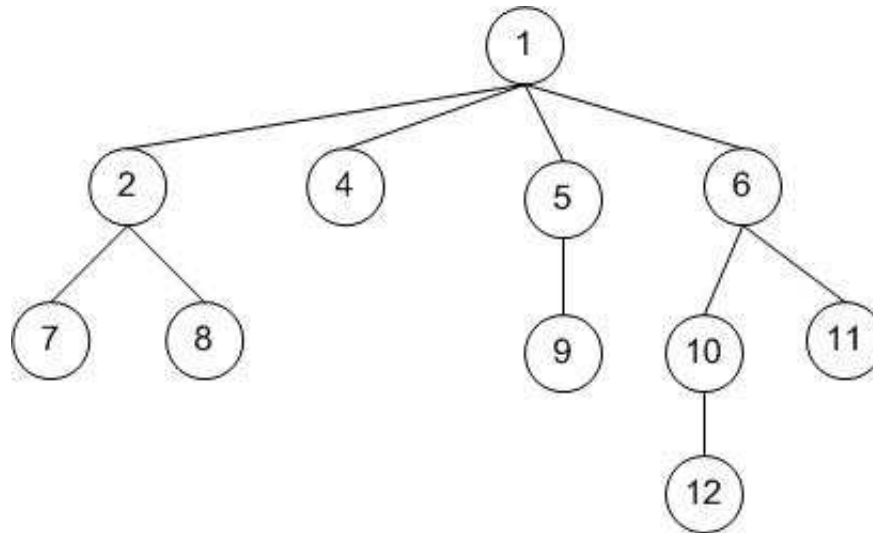
Implementation of Binary Tree

Binary Tree: a tree in which no node can have more than two children.



```
class BinaryNode
{
    Object element;
    BinaryNode left;
    BinaryNode right;
}
```

Implementation of Trees

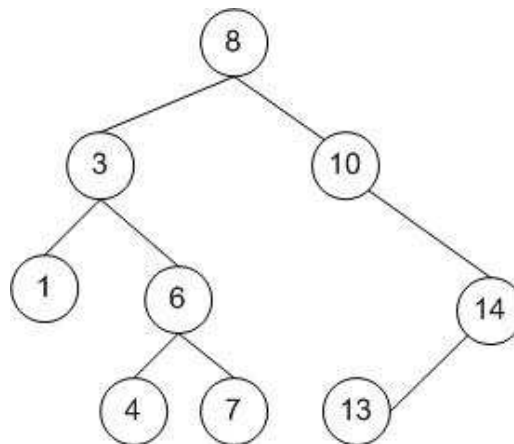


```
class TreeNode
{
    Object element;
    TreeNode firstChild;
    TreeNode nextSibling;
}
```

Binary Search Tree

Binary Search Tree: a binary tree that has the following property:

for every node, X , in the tree, the values of all the items in its left subtree are smaller than the item in X , and the values of all the items in its right subtree are larger than the item in X .



Question: How to print all the items in a binary search tree in order?

Tree Traversal (recursive, using stack)

preorder traversal: work at a node is performed before its children are processed.

```
public void PreorderTraversal(TreeNode T)
{
    TreeNode sibling;
    //if the tree is empty, we're done
    if(T==null) return;

    //work on the node
    visit(T.element);

    //work on subtrees
    PreorderTraversal(T.firstChild);

    //work on sibling trees
    PreorderTraversal(T.nextSibling);
}
```

See example.

Applications: see P.162, 4.32

Design a recursive linear-time algorithm that tests whether a binary tree satisfies the search tree order property at every node. (Solution A)

Tree Traversal (Postorder traversal)

postorder traversal: work at a node is performed **after its children are processed.**

```
public void PostorderTraversal(TreeNode T)
{
    TreeNode sibling;
    //if the tree is empty, we're done
    if(T==null) return;

    //work on subtrees
    PostorderTraversal(T.firstChild);

    //work on sibling trees
    PostorderTraversal(T.nextSibling);

    //work on the node
    visit(T.element);
}
```

See example.

Applications:

1. find the height of a tree;
2. count the total number of leaves in a tree, etc.

Inorder Traversal for Binary Tree(recursively, using stack)

inorder traversal: process left subtree, followed by the node, then the right subtree.

```
public void InorderTraversal(BinaryNode T)
{
    //if the tree is empty, we're done
    if(T==null) return;

    //work on left subtree
    InorderTraversal(T.left);

    //work on the node
    visit(T.element);

    //work on right subtree
    InorderTraversal(T.right);
}
```

Applications:see P.162, 4.32

Design a recursive linear-time algorithm that tests whether a binary tree satisfies the search tree order property at every node. (Solution B)

How to do it? Compare with Solution A. Which one is more efficient?

Level-order Traversal (iteratively, using queue)

P.145 level-order traversal: all nodes at depth d are processed before any node at depth $d + 1$.

Solution to Quiz3 (By Justin)

<http://cs.uiowa.edu/jhffmn/CS21/>